LABORATORIJSKA VEŽBA BR. 7

Primena tehnike digitalnog potpisa

- Upoznavanje sa tehnikom digitalnog potpisa
- Primena heš funkcije MD5
- Testiranje primene digitalnog potpisa

POTREBNA OPREMA

- Računar sa instaliranim Windows operativnim sistemom
- Instalirani programski paket Cryptool

TEORIJSKE OSNOVE

Tehnika digitalnog potpisa

Tehnika digitalnog potpisa koristi tehniku asimetričnog kriptovanja. Pošiljalac i primalac imaju par ključeva od kojih je jedan tajni, a drugi svima dostupan javni ključ. Ključevi predstavljaju matematičke algoritme koje je izdalo sertifikaciono telo. Digitalni potpisi se koriste za identifikaciju izvora informacije, što može biti neka osoba, organizacija ili računar. Sama ideja digitalnog potpisa slična je klasičnom potpisivanju dokumenata jer, ukoliko se neki dokument želi poslati elektronskim putem, on se mora i potpisati, pri čemu je, za razliku od klasičnog potpisa, digitalni potpis gotovo nemoguće falsifikovati. Na osnovu iznetog se zaključuje da je za funkcionalnost digitalnog potpisa potrebno izvršiti dva procesa, od kojih jedan sprovodi potpisnik, a drugi primalac. Uspešnom proverom digitalnog potpisa garantuju se:

1. ***autentičnost*** – pouzdanost identiteta pošiljaoca posledica je činjenice da je otisak poruke koji je šifrovan tajnim ključem moguće uspešno dešifrovati samo primenom odgovarajućeg javnog ključa;
2. ***integritet*** – upoređivanjem izračunatog i dešifrovanog otiska poruke utvrđuje se da poruka nije modifikovana;
3. ***neporicivost*** – pošiljalac ne može da porekne slanje poruke pošto je potpisana njegovim tajnim ključem.

Važno je napomenuti da elektronski potpis uopšte, pa tako ni digitalni potpis, ne pruža zaštitu tajnosti podataka od neovlašćenog čitanja jer se svi podaci šalju u svom originalnom (nepromenjenom) obliku.

Digitalni sertifikat

Kreiranje digitalnog potpisa i njegova verifikacija vrše se, kako je već pomenuto, asimetričnim kriptografskim sistemima prilikom čega se koriste: tajni (privatni) ključ poznat jedino korisniku i javni ključ poznat širem krugu ljudi, a ne samo primaocu. Postavlja se pitanje kako možemo biti sigurni da je to zaista javni ključ potpisnika. Rešenje ovog problema postiže se upotrebom digitalnog sertifikata. Digitalni sertifikat je digitalno potpisani dokument koji povezuje javni ključ s osobom kojoj pripada i možemo ga nazvati i digitalnom ličnom kartom jer on to i zaista jeste, tj. digitalna lična karta u „cyber prostoru“, sredstvo kojim dokazujemo identitet na Internetu.

Digitalni sertifikat (*digital certificate*) predstavlja element kojim se utvrđuje veza između identiteta subjekta i njegovog javnog ključa primenom asimetričnog algoritma. Elementi koji čine strukturu digitalnog sertifikata su: verzija formata sertifikata, serijski broj sertifikata, identifikator algoritma, naziv sertifikacionog tela, rok važnosti sertifikata, vlasnik sertifikata, polje dodatnih atributa, informacija o

javnom ključu vlasnika i digitalni potpis sertifikata od strane ustanove koja je izdala sertifikat (SA od početnih engleskih reči *certificate authority*).
Prema dosadašnjim iskustvima ovakva struktura sertifikata ispunjava zahteve savremenih kriptografskih sistema zaštite. Većina savremenih sistema zaštite, koji uključuju infrastrukturu sa javnim ključevima (PKI od početnih engleskih reči *public key infrastructure*) bazira se na primeni digitalnog sertifikata.

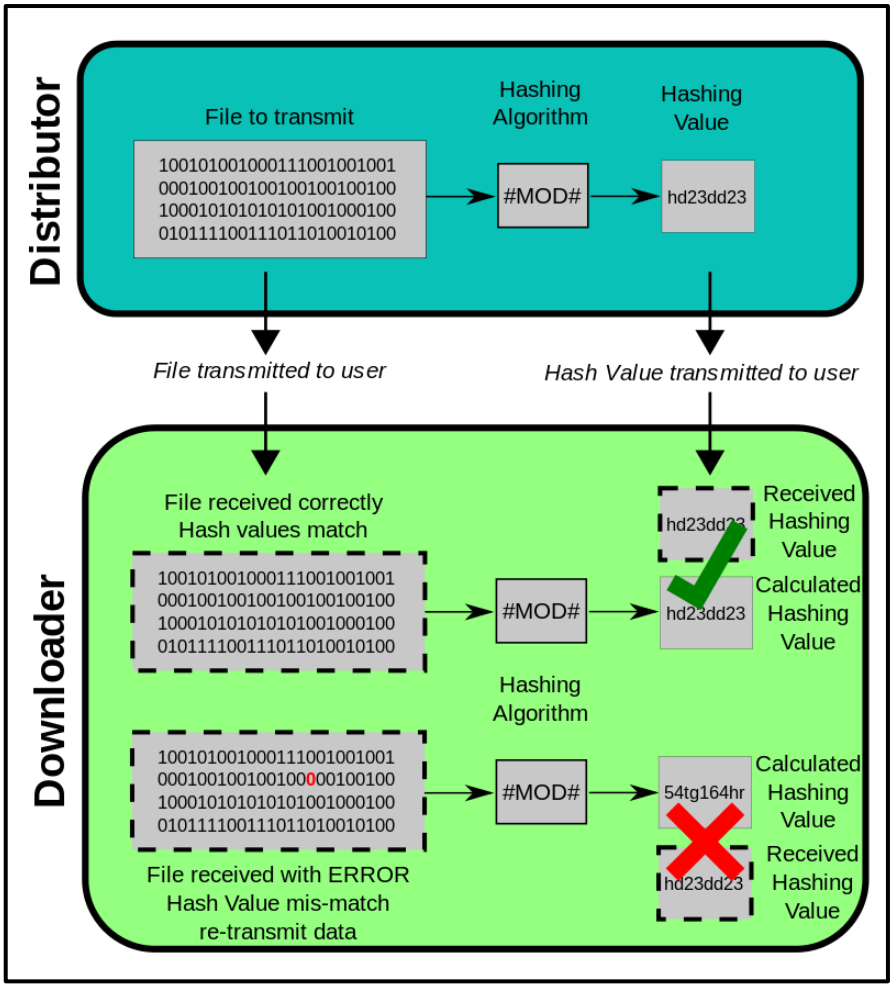
Potpisivanje sažetka poruke

Ne štite sve vrste opisanih algoritama šifrovanja integritet, odnosno verodostojnost poruke koja je šifrovana. Funkcija za sažimanje (skraćivanje, kompresiju...) ili heš (*hash*) funkcija jeste tehnika koja obezbeđuje proveru integriteta poruke, što je važno jer je moguće da je ključ otkriven i da nam napadač šalje lažne poruke, ali i da je došlo i do greške prilikom šifrovanja, tako da primljena poruka nije identična originalnom dokumentu. Iz tog razloga kreirane su pomenute heš ili funkcije za sažimanje, koje se mogu sresti i pod imenima *one-way*, *hash function*, *message digest*, *fingerprint* algoritmi. Najpoznatiji i najkorišćeniji heš algoritmi su: *SHA-224, SHA-256, SHA-384, SHA-512, MDC-2, RIPEMD-160*, kao i stariji *SHA-1* (*Secure Hash Algorithm 1*) sa 160-bitnim sadržajem, *MD5* (*Message Digest 5*) sa 128-bitnim trebalo bi da budu povučeni iz upotrebe. Heš algoritmi se svrstavaju u kriptografske algoritme bez ključa.

Kao što smo u prošloj vežbi videli, ovi heš algoritmi prosto sažmu (u bukvalnom prevodu samelju) svaku poruku ili fajl bez obzira na veličinu i na izlazu dobijamo poruku konstantne dužine, u zavisnosti od algoritma. Iz dobijenog izlaza nemoguće je rekonstruisati ulaznu poruku, a bitno je da je, isto tako, gotovo nemoguće kreirati dve smislene poruke koje će imati iste vrednosti heš funkcije, te tako mi u svakom trenutku možemo da proverimo integritet poruka, odnosno da primetimo razliku u tekstu primljene poruke, prostim ponovnim proračunavanjem heš funkcije i upoređivanjem dobijenih rezultata. Verovatnoća da u poruci neko izmeni neku stavku, tako da novi dobijeni tekst ima istu heš vrednost kao i originalni tekst, u slučaju 160 bitnih algoritama jeste zanemarljiva, zato se negde heš funkcije nazivaju i otisci prstiju poruka. Ukoliko su poruke duge, korišćenje kriptovanja sa javnim ključem za potpisivanje cele poruke veoma je nepraktično zbog velikih dužina poruka, što traži dosta resursa a i troši mnogo vremena za kriptovanje. Zato se kao logično rešenje ovog problema javlja mogućnost, potpisivanja samo sažetka umesto potpisivanja cele poruke. Osoba koja šalje poruku kreira skraćenu verziju poruke tj. njen sažetak. Tako formiran sažetak potpisuje i šalje komunikacionim kanalom a osoba koja primi tako skraćenu poruku proverava njen potpis. Svaka promena izvorne poruke izaziva promenu u sadržaju, što se odražava na promenu potpisa, čime se minimizuje mogućnost zloupotrebe.
Tehnika potpisivanja sadržaja poruke uglavnom koristi neku od dve heš funkcije: *MD5* (*Message Digest 5*) sa 128-bitnim sadržajem i *SHA-1* (*Secure Hash Algorithm 1*) sa 160-bitnim sadržajem. Za garantovanje sigurnosti poruke heš funkcija mora zadovoljiti dve stvari:

1) **funkcija sažimanja se obavlja u jednom smeru** - sadržaj se jedino može formirati na osnovu originalne poruke, ali ne i obrnuto a formiranje sadržaja treba da bude brzo i jednostavno;
2) **heš funkcija je jednoznačna**, tj. primena iste heš funkcije na istoj poruci daje isti sažetak.

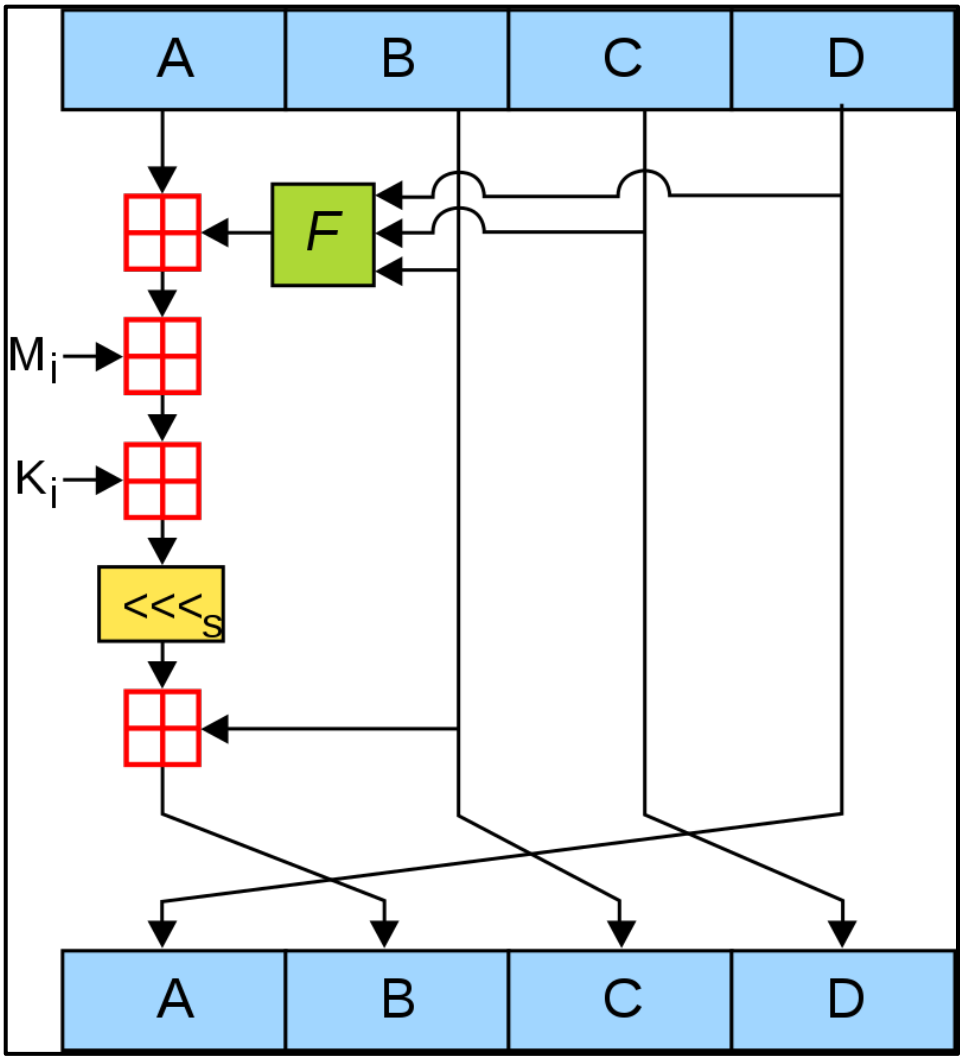
Nakon kreiranja sadržaja poruke, vrši se kriptovanje (potpisivanje) iste korišćenjem tajnog ključa osobe koja šalje poruku (***Distributor***). Obično se za kriptovanje koristi *RSA* algoritam. Kriptovani sadržaj se upakovan zajedno sa originalnom porukom šalje ***Downloader***-u. ***Distributor*** primenom heš funkcije formira sažetak koji se potpisuje i upakovan sa porukom šalje ***Downloader***-u. ***Downloader*** prima originalnu poruku i kriptovani sadržaj zajedno sa potpisom, pa nakon prijema vrši razdvajanje na poruku i potpis. Primenom heš funkcije na originalnu poruku ***Downloader*** kreira drugi sadržaj tj. formira svoju heš vrednost. Takođe, dekriptuje kriptovani sadržaj koji je primila od ***Distributor***-a uz pomoć javnog ključa ***Distributor***-a. Konačno, vrši poređenje primljenog digitalnog potpisa sa novokreiranom heš vrednošću koju je on dobio i ukoliko su te vrednosti iste verifikacija poruke je uspešno obavljena. Proces kreiranja i provere digitalnog potpisa prikazan je na donjoj slici.



Svrha digitalnog potpisa je da potvrdi autentičnost sadržaja poruke (dokaz da poruka nije promenjena na putu od pošiljaoca do primaoca), kao i da obezbedi garantovanje identiteta pošiljaoca poruke. Pomoću svog potpisa korisnik ovlašćuje neku radnju i preuzima odgovornost za nju. *MD5* (*Message Digest Algorithm 5*) jeste heš funkcija koja se primenjuje u aplikacijama za digitalno potpisivanje dokumenata. Dužina sadržaja koji se formira na osnovu *MD5* funkcije je kratka (128 bita) što ga čini pogodnim za brzu proveru identiteta osoba koje šalju obimne dokumente. Algoritam *MD5* funkcije je razvio Ron Rivest (Ron Rivest) 1991. godine, kao zamenu za *MD4* algoritam. Nakon pet godina otkriveni su mali nedostaci u algoritmu, te su kriptografi preporučivali upotrebu drugih heš funkcija. Nekoliko narednih godina su otkriveni dodatni nedostaci pa je upotreba ovog algoritma dovedena u pitanje. Tokom 2005. godine grupa istraživača je uspela da formira isti sadržaj primenjujući *MD5* na dva različita dokumenta. Zbog pronađenih nedostataka, danas se ovaj algoritam sve ređe koristi za digitalno potpisivanje, ali je našao primenu u proveri integriteta fajlova, gde se koristi za izračunavanje kontrolnih suma, kod kojih sigurnost nije prioritetna. *MD5* algoritam kao ulaznu informaciju koristi *w*-bitni broj. Izvorni tekst se može prikazati kao niz brojeva:

$$\mathbf{m_0, m_1, m_2,\ldots,m_{w-2}, m_{w-1}}$$

gde je broj *w*, vrednost iz proširenog skupa prirodnih brojeva.
Na početku je potrebno izvršiti dopunu ulazne informacije do vrednosti koja se dobija od broja koji je celobrojni umnožak od 512 bita umanjenog za 64 bita. Na primer, ukoliko se izvorna poruka sastoji od 128 bitova ($w = 127$) potrebno je dopuniti je do 448 bitova, tj. 512 – 64 = 448. Dopuna se započinje sa početnim bitom „1“, a svi ostali bitovi za popunjavanje imaju vrednost „0“. Nakon dopune poruke, izvornoj poruci je potrebno dodati 64-bitnu reprezentaciju broja *w*. Ukoliko je dužina poruke veća i ne može da se predstavi pomoću 64 bita, poruci se dodaje samo nižih 64 bita. Dodavanjem ovih 64 bita dužina cele poruke postaje deljiva sa 512, odnosno deljiva sa 16 reči od 32- bita. Sada se poruka može prikazati kao: **M[1,2,..., N]** gde je *N* broj deljiv sa 16. Ovako pripremljenu poruku algoritam kasnije koristi prilikom formiranja sadržaja. Na donjoj slici prikazan je izgled jednog od moguća 64 koraka izvršenja *MD5* algoritma.



Nakon predhodne pripreme poruke, potrebno je inicijalizovati 128-bitni bafer koji se sastoji od četiri 32-bitna registra A, B, C i D. Kao inicijalne vrednosti koje se upisuju u ove registre koriste se proizvoljne 32- bitne konstante. Kada se završi inicijalizacija, pokreće se prvi korak MD5 algoritma. Ukupnih 64 koraka se deli u četiri ciklusa od po 16 koraka. Algoritam je formiran za izvršenje 512 bita poruke, što znači da ukoliko je poruka duža od 512 bita izvršenje algoritma se mora ponoviti. Algoritam se sastoji od četiri ciklusa koji imaju isti tok s tim što se prilikom izračunavanja u svakom od ciklusa koristi različita logička funkcija F, G, H i I. Funkcije se računaju po formulama:

F(X,Y,Z)=(X¬ AND Y) OR (NOT X AND Z)
G(X,Y,Z)=(X AND Z) OR (Y AND NOT Z)
H(X,Y,Z)=X XOR Y XOR Z
I(X,Y,Z)=Y XOR (X OR NOT Z)

Gde su AND, OR, NOT i XOR matematičke logičke operacije.
Tokom ciklusa se koriste operacije aritmetičkog sabiranja po modulu 2^{32} i operacija pomeranja ulevo za S pozicija, gdje je S vrednost različita za svaki ciklus. M[N] predstavlja 32-bitnu ulaznu poruku, a K[N] je konstanta koja je drugačija za svaki ciklus. Ukupno 16 M[N]-ova se koristi tokom 16 koraka u okviru svakog od 4 ciklusa. Rezultat jednog koraka se koristi kao početna vrijednost (*A*, *B*, *C i D*) narednog koraka. Na kraju se konačna vrednost formiranog sadržaja upisuje u registre *A*, *B*, *C* i *D*.

ZADATAK:

1. **Demonstracija na Cryptool-u**

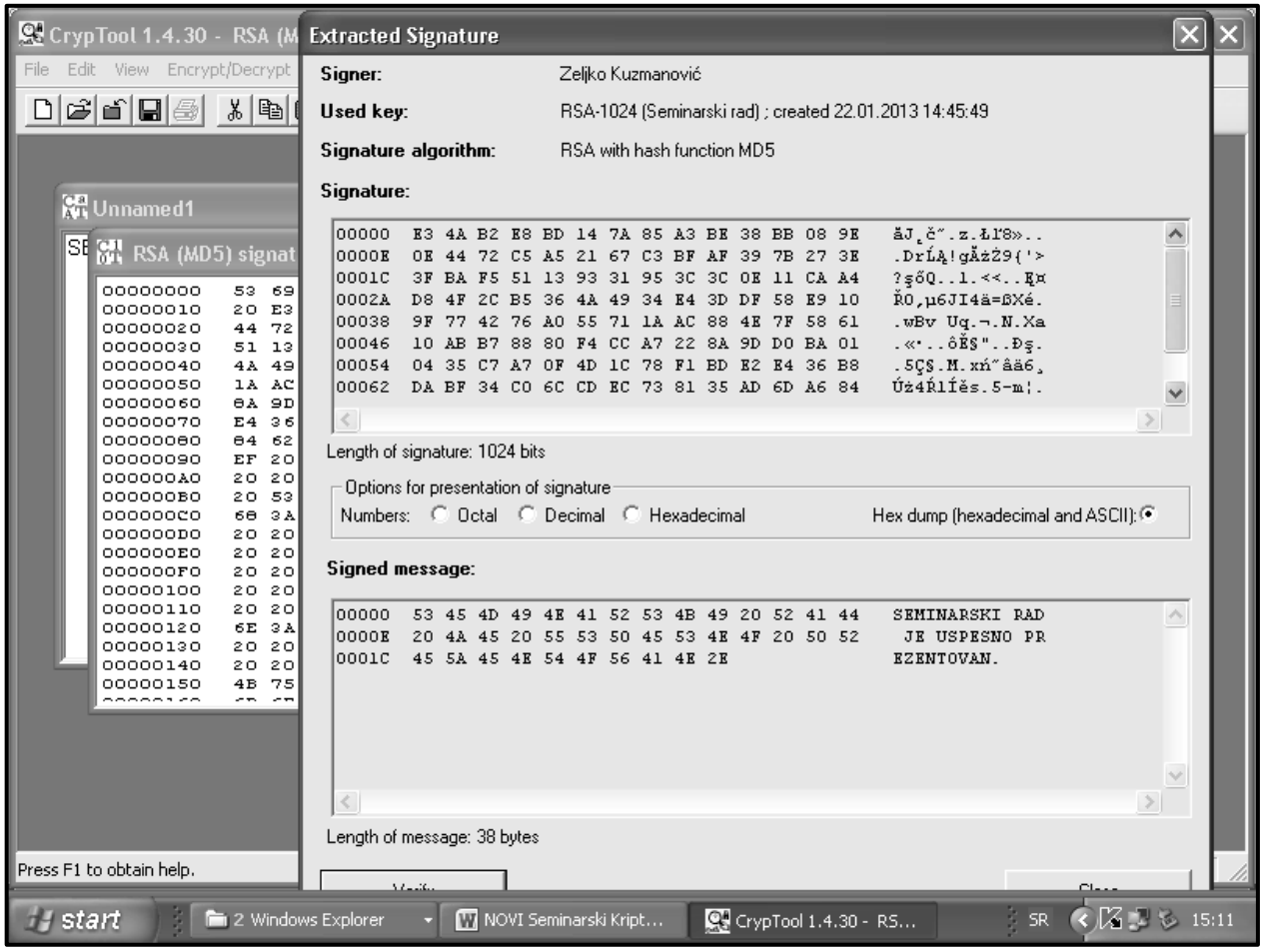
Startujemo program *CrypTool*. U glavnom meniju idemo na opciju ***File***, i izaberemo ***New*** i otvorimo novi prozor, ***Unnamed1***, u koji upisujemo neku našu poruku ili tekst koji želimo da zaštitimo (kriptujemo). Kao primer za našu vežbu unećemo sledeću poruku:

SEMINARSKI RAD JE USPEŠNO PREZENTOVAN.

Posle upisivanja poruke idemo na opciju ***Digital Signatures/PKI***, pa zatim izaberemo opciju ***Sign Document***. Otvara se novi prozor u kome treba izabrati nekoliko stvari kao što su:

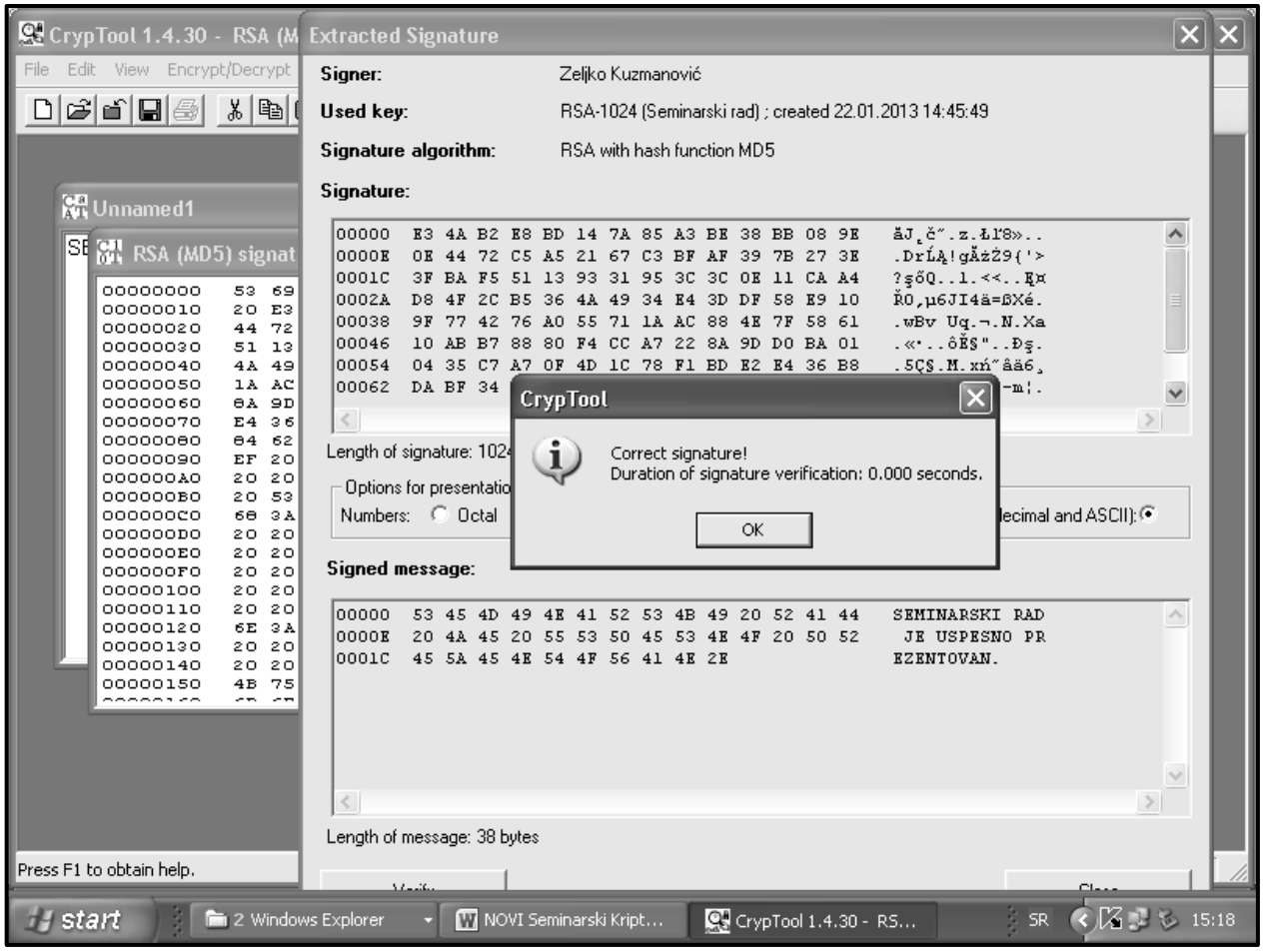
1) heš funkcija sa kojom želimo da radimo,
2) algoritam koji želimo da koristimo za digitalni potpis,
3) ključ koji želimo da koristimo za potpis,
4) PIN kod za izabrani ključ.

Nakon što smo selektovali sve tražene opcije i upišemo i PIN kod (unesite PIN kod 1111), te pritiskom na taster ***Sign***, otvara se novi prozor u kom se nalazi potpisani sažetak naše poruke, kao i vreme koje je bilo potrebno da se kreira taj sažetak. Posle ovoga vršimo verifikaciju potpisanog sažetka poruke (potpisa) koji se inače obavlja na prijemnoj strani, tj. vrši se od strane osobe kojoj šaljemo poruku. U glavnom meniju idemo na opciju ***Digital Signatures/PKI*****,** pa izaberemo opciju ***Extract Signature***. Otvara se novi prozor u kome imamo osnovne podatke o osobi koja je potpisala poruku, ključu, korišćenom algoritmu, potpisanoj poruci kao i potpisu (kaon a prikazanoj slici):



Pritiskom na taster ***Verify***, otvara se novi prozor ***Signature Verification*** u kome se traži da izaberemo potpis osobe koja je poslala prvobitnu poruku. Neka je ta osoba:

Na prozoru pored ovoga imamo i određene informacije koje se odnose na postupak verifikacije digitalnog potpisa koji je u toku. Pritiskom na taster ***Verify signature***, izvršena je verifikacija potpisa, kao i osobe koja je potpisala sadržaj poruke koja je prikazana na donjoj slici.



Pitanje: Kako izgleda digitalni potpis koji ste dobili ako se potpisuje sledeća poruka:

Student (vaše ime i prezime) **je odradiio uspešno Lab. vežbu broj 6.**

Zadatak: Odraditi digitalni potpis za pet primera po izboru studenta.